

# Utiliser le serveur

Ce serveur est fait pour nous, les étudiants ! Que ce soit pour des projets perso ou pour un club, tu peux utiliser le serveur Etud pour mettre en ligne un site web ou partager des fichiers gratuitement.

- Règles d'utilisation d'Etud
- Technologies disponibles
- Espace web étudiant
  - Pour les clubs
  - Pour les étudiants
  - Créer une base de donnée
- Accès au serveur
  - Accès par SSH
  - Accès par gestionnaire de fichiers
  - Accéder à sa base de donnée
- Utilisation du serveur
  - Structure de votre espace web
  - Créer votre site web
- Systèmes d'authentications
  - Utiliser le CAS (Recommandé)
  - Utiliser un .htaccess
  - Utiliser le LDAP
- Astuces
  - Changer son mot de passe

# Règles d'utilisation d'Etud

Le serveur Etud est maintenu par les Roots Étudiants, des bénévoles travaillant dans leur temps libre, en plus des cours. Nous comptons donc sur votre compréhension quant à la vitesse de réponse lors de vos demandes. Pour simplifier la vie de tout le monde, les Roots Étudiants ont décidé d'imposer les règles d'utilisation suivantes, que **tout étudiant et club se doit de respecter sous peine de vous voir exclure du serveur étud.**

## Stockage

Le serveur étud est partagé entre tous les étudiants et possède des ressources limitées, nous vous remercions donc de respecter ce partage et de ne pas en abuser. Nous n'imposons pas de réelle limite sur la quantité et la taille des fichiers que vous pouvez stocker sur votre espace club/étudiant. Cependant **si vous voulez stocker plus de 100-200Mo, merci de contacter les Roots Étudiants ([root-etud@insa-toulouse.fr](mailto:root-etud@insa-toulouse.fr)) pour les prévenir.**

## Gestion de comptes externes

Pour des raisons de sécurité, il est **interdit de stocker des mots de passe/identifiants sur le serveur étudiant.** De ce fait, il est **interdit de créer votre propre système de création/gestion de comptes.** Si vous souhaitez implémenter une authentification utilisateur, vous devrez utiliser les systèmes d'authentification fournis par l'INSA, (Le LDAP de l'INSA ou le CAS ). Ces méthodes utilisent l'annuaire INSA et donc limitent la connexion aux utilisateurs ayant un compte INSA, sans avoir à stocker des mots de passe sur le serveur étud.

**Si vous souhaitez tout même créer votre propre système de gestion de compte, vous devrez utiliser un autre serveur.**

La seule exception à cette règle est l'utilisation de comptes administrateurs pour la gestion de votre site (accès à la console wordpress ...) . Dans ce cas, merci de contacter les Roots Étudiants pour en discuter.

## Données personnelles

Il est interdit de stocker des données personnelles sur le serveur étudiant, telles que, mais non limitées à :

- Les mots de passe (même hashés)
- Les adresses email (insa ou personnelles) \*
- Les numéros de téléphone \*
- Les adresses postales \*

*\* Font exception aux points précédents les informations ayant pour objectif leur diffusion publique (e.g. CV, mail de contact, ...). La diffusion de ces informations est sous la responsabilité de l'étudiant. Vérifiez bien d'avoir le droit de publier ces informations si elles ne vous appartiennent pas.*

Il est possible que ces données se retrouvent sans protection suite à des failles de sécurité. Suite à de trop nombreux abus et incidents de sécurité, les Roots Étudiants ont pris la décision d'interdire le stockage de ces données sur le serveur étud.

# Technologies disponibles

Le site que vous voulez créer dépend des technologies disponibles sur le serveur Etud. Si la technologie dont vous avez besoin n'apparaît pas dans la liste ci-dessous, merci de contacter les Roots Étudiants ([root-etud@insa-toulouse.fr](mailto:root-etud@insa-toulouse.fr)).

## Technologies disponibles

- **PHP** 7.3 avec support pour les `.htaccess`
- **Python** 2.7, 3.5 et 3.6 avec support pour les `venv`
- **MySQL** (usage restreint)
- *Il y en a beaucoup en fait, n'hésitez pas à aller simplement [vérifier sur le serveur...](#)*

# Espace web étudiant

Comment faire pour avoir un espace web ?

Espace web étudiant

# Pour les clubs

Les clubs ont le droit d'avoir un espace web, avec aussi une base de donnée si ils le souhaitent.

## Créer son espace

**Avant de continuer, merci de bien lire les Règles d'utilisation d'Etud.**

Contrairement aux espaces personnels, les espaces clubs ne peuvent pas être créés automatiquement. Il faut pour cela contacter les Roots Étudiants par mail ([root-etud@insa-toulouse.fr](mailto:root-etud@insa-toulouse.fr)).

Si vous voulez bénéficier d'une base de donnée, merci de regarder sur [cette page](#).

# Pour les étudiants

Les étudiants ont le droit d'avoir un espace web, avec aussi une base de donnée s'ils le souhaitent.

## Prérequis

**Avant de continuer, merci de bien lire les Règles d'utilisation d'Etud.**

Pour pouvoir accéder à son espace web étudiant, il faut :

- Être étudiant à l'INSA
- Être connecté sur le VPN

## Création de l'espace

Si vous êtes étudiant, et que vous n'avez toujours pas activé votre espace. Sachez qu'il suffit de se connecter une fois au serveur `etud.insa-toulouse.fr` afin de le créer.

Pour cela ouvrez un terminal sur votre ordinateur (Mac, Linux ou Windows) et tapez la commande suivante :

```
ssh login@etud.insa-toulouse.fr
```

En remplaçant `login` par votre login INSA. Par exemple `"ssh michel@etud.insa-toulouse.fr"` pour l'utilisateur avec le login `"michel"`.

Vous aurez sûrement un message vous demandant de confirmer la connexion, tapez `"yes"` et puis appuyez sur la touche Entrée.

Attendez un peu puis entrez votre mot de passe INSA. Rien ne s'écrit quand vous tapez c'est normal. Appuyez sur la touche Entrée quand vous avez terminé de taper votre mot de passe puis attendez que la connexion s'établisse.

Une fois que vous avez une ligne similaire à l'image ci-dessous, tapez `"exit"`, puis appuyez sur la touche Entrée pour quitter.

```
Last login: Wed Mar 18 16:07:51 2020 from 10.29.40.24
vergnet@srv-falcon:~$ █
```

Votre espace est maintenant créé, vous pouvez fermer votre console.

## Envoyer des fichiers

Plusieurs méthodes sont possibles pour envoyer vos fichiers (ftp, filezilla, etc.), ici on vous en donne une qui fonctionnera sur tous les ordinateurs : SFTP.

- Connectez-vous au VPN
- Sur votre machine locale ouvrez votre meilleur terminal (Windows : Super+x puis *Windows Powershell*, Linux : Ctrl+Alt+T ou rechercher "Terminal")
- Tapez `sftp login@etud.insa-toulouse.fr`
- Si on vous demande la confiance, tapez `yes`
- Entrez votre mot de passe
- Et voilà, vous êtes connecté ! Voici les commandes qui vous seront utiles :
  - `ls` pour lister les fichiers et répertoires
  - `cd` pour changer de répertoire (`cd ..` pour remonter)
  - `get -r file .` pour récupérer un fichier
  - `get -r * .` pour récupérer tous les fichiers d'un répertoire
  - `put -r file .` pour mettre un fichier en ligne

## Base de données MySQL

Chaque étudiant à accès à une base de données MySQL.

Cette base de donnée n'est pas créée automatiquement en même temps que l'espace web, car tous les étudiants n'en ont pas besoin.

Pour pouvoir y accéder, merci de faire une demande aux Roots étudiants à l'adresse root-etud@insa-toulouse.fr.

### Accès via navigateur (phpMyAdmin)

Cette base de donnée est accessible par navigateur à l'adresse : <https://www.etud.insa-toulouse.fr/phpmyadmin/> phpMyAdmin.

Si vous obtenez des erreurs du type "Error: Token mismatch", tentez de changer de navigateur et/ou de vider le cache de votre navigateurs et supprimer les cookies (Ctrl+Shift+Suppr).

# Accès via le terminal (ssh)

Si cela ne marche toujours pas, ce qui est probablement le cas, vous allez devoir utiliser votre terminal (mais pas d'inquiétude ça reste simple)

- Connectez-vous au VPN
- Connectez vous en ssh depuis votre terminal
- Utilisez la commande mysql (utilisez le manuel pour connaître les possibilités)

Par exemple, pour importer une table exportée en local (remplacez utilisateur par votre login) :

```
ssh utilisateur@etud.insa-toulouse.fr
mysql -u utilisateur -p utilisateur < path/to/exportedtable.sql
```

et pour y accéder via votre page php :

```
$db = new PDO("mysql:host=localhost;dbname=user;charset=utf8", 'leNomDeLaTableImportée',
'leMotDePasseFourniAuRootEtudiant');
```

# Problèmes courants

## 403 Forbidden - Server unable to read htaccess file

Si vous avez ce message d'erreur en accédant à votre page web : 403 forbidden.

Cela signifie que les droits d'exécution (""+x"") ne sont pas activés sur votre home ("~") et/ou sur le répertoire "~/public\_html/". Pour solutioner le problème, connectez vous en SSH au serveur et tapez :

```
chmod +x ~ ~/public_html/
```

Ou bien dans Filezilla : clic droit sur votre home -> droits d'accès au fichier -> mettre 711 dans le champ "valeur numérique". Effectuez la même opération sur votre dossier public\_html.

# Créer une base de donnée

Les étudiants et les clubs peuvent, en plus de leur espace web, avoir une base de donnée sur Etud. Cette page explique comment la créer et y avoir accès.

## Demander la création

La création n'est pas automatique avec l'espace club/étudiant, car tout le monde n'en a pas l'utilité. Pour des raisons de sécurité, il faut demander la création de cette base de donnée aux Roots Étudiants ([root-etud@insa-toulouse.fr](mailto:root-etud@insa-toulouse.fr)). Lors de cette demande par mail, **il est obligatoire** d'expliquer pourquoi la base de donnée est nécessaire et quelles données vont être stockées.

**Si l'utilisation décrite ne respecte pas les Règles d'utilisation d'Etud**, les Roots Étudiants se réservent le droit de refuser la création de la base de donnée.

**Si le type de données stockées change**, il est obligatoire de notifier les Roots, qui peuvent refuser ces changements, voir supprimer la base de donnée si le changement ne respecte pas les **Règles d'utilisation d'Etud**.

## Création

Si votre demande est acceptée, un Root Étudiant vous répondra avec vos identifiants. Vous pourrez alors suivre [ce guide pour accéder à votre base de donnée](#).

# Accès au serveur

Comment accéder à son espace web une fois créé ?

# Accès par SSH

Le SSH est une méthode de connexion par le terminal. Mais ne vous inquiétez pas ! Ce tutoriel va vous guider pas à pas, **aucune connaissance en informatique est nécessaire**.

## Prérequis

- Être étudiant INSA
- Être connecté au VPN

## Connexion

Pour commencer, ouvrez un terminal sur votre ordinateur (Mac, Linux ou Windows 10) et tapez la commande suivante :

```
ssh login@etud.insa-toulouse.fr
```

En remplaçant `login` par votre login INSA. Par exemple `ssh michel@etud.insa-toulouse.fr` pour l'utilisateur avec le login `michel`.

Vous aurez sûrement un message vous demandant de confirmer la connexion, tapez `yes` et puis appuyez sur la touche `Entrée`.

Attendez un peu puis entrez votre mot de passe INSA. **Rien ne s'écrit quand vous tapez c'est normal**. Appuyez sur la touche `Entrée` quand vous avez terminé de taper votre mot de passe puis attendez que la connexion s'établisse.

Une fois que vous avez une ligne similaire à l'image ci-dessous, cela veut dire que vous avez réussi à vous connecter ! Si vous aviez suivi ce guide pour [créer votre espace web étudiant](#), bravo vous avez réussi !

SshEtudExemple.png

Si vous êtes aisé avec le terminal, vous pouvez naviguer sur votre espace comme vous le souhaitez.

Si vous avez peur du terminal, vous pouvez suivre la suite de ce guide pour vous déconnecter, puis lisez [comment utiliser un logiciel plus simple](#).

# Déconnexion

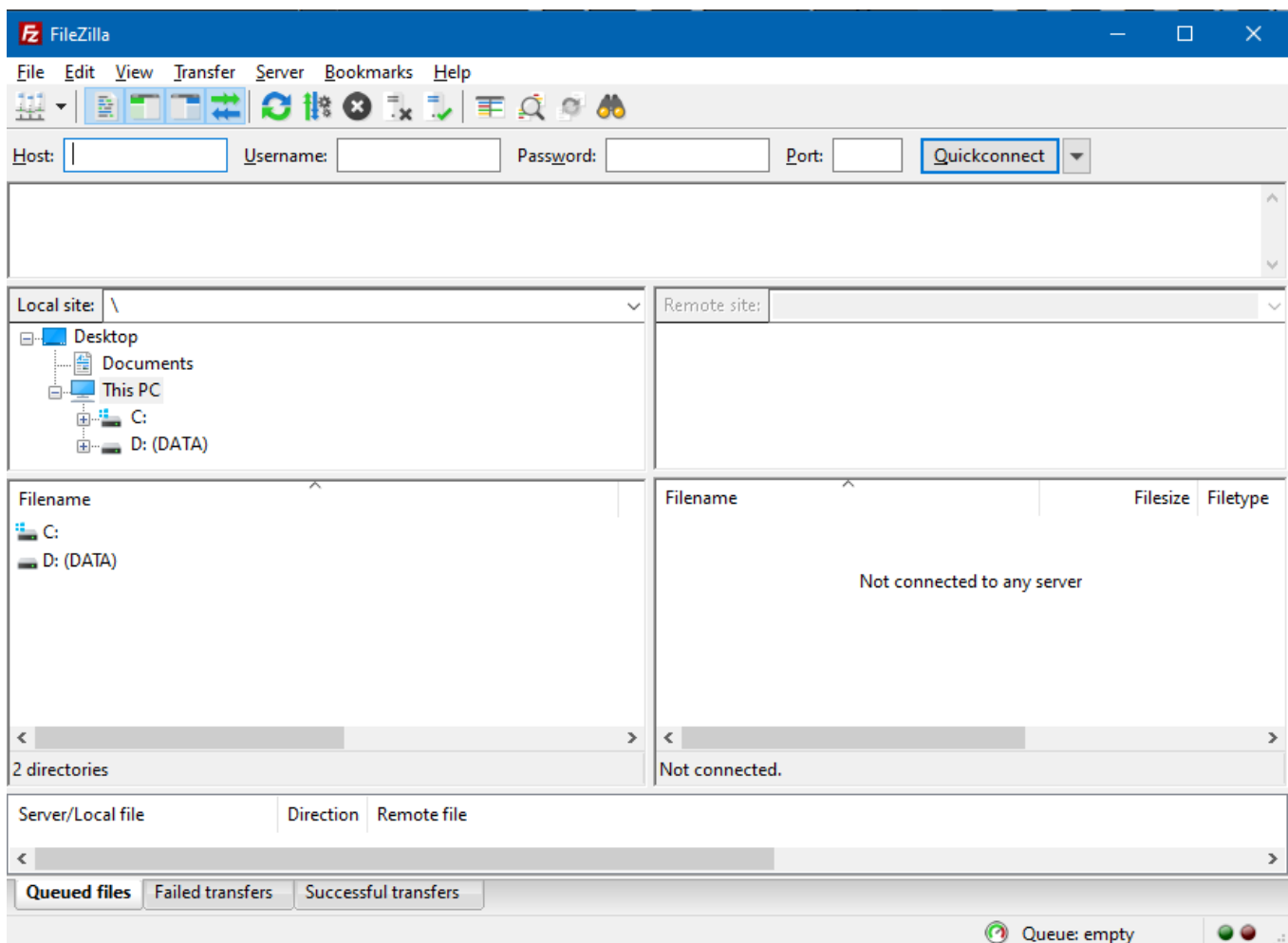
Pour vous déconnecter du serveur, tapez tout simplement la commande `exit` dans le terminal.

# Accès par gestionnaire de fichiers

L'accès par gestionnaire de fichiers est sans doute la manière la plus simple de déposer ou récupérer un fichier sur le serveur Etud. Il vous **faut cependant être connecté au VPN**.

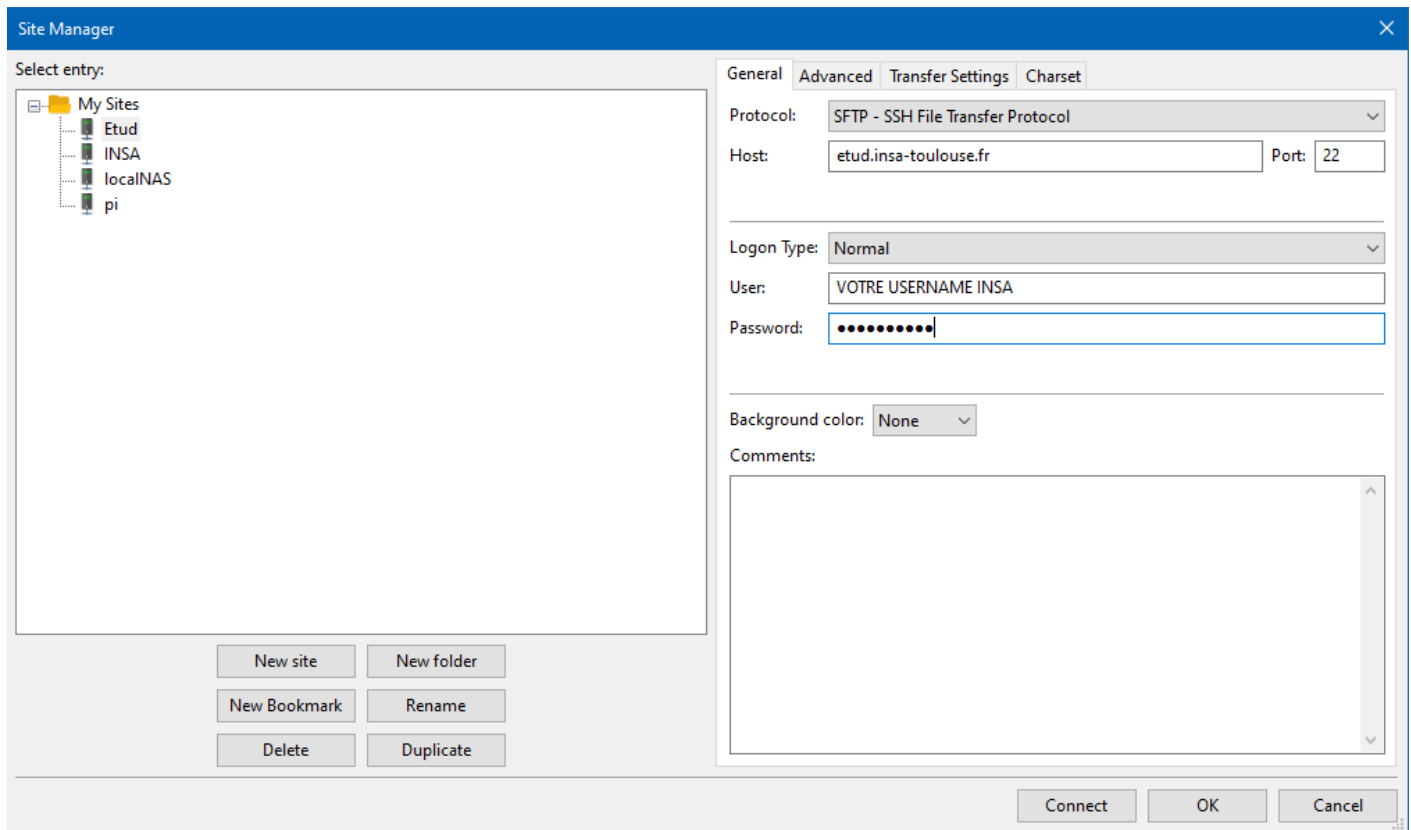
## Windows

Sous Windows, la solution la plus simple reste de télécharger [Filezilla Client](#). Il est important de choisir Filezilla client et non pas Filezilla serveur. Une fois téléchargé, installé et ouvert vous aurez la page suivante :



Vous remarquerez que la zone de gauche correspond à la structure de votre PC. Vous pouvez simplement naviguer jusqu'au dossier souhaité.

Il vous faut maintenant vous connecter au serveur Etud. Premièrement, vérifiez que vous êtes connecté au VPN de l'INSA. Ensuite vous pouvez ouvrir le "Site Manager" (**File > Site Manager**), cliquer sur "New Site" et le remplir comme l'exemple ci dessous :



Les points importants sont :

- Protocol : **SFTP**
- Host : **etud.insa-toulouse.fr**
- Port : **22**
- User : **Votre username INSA (ne tapez pas "VOTR USERNAME INSA")**
- Password : **Votre mot de passe INSA (ne tapez pas "●●●●●●●●●●")**

Vous pouvez ensuite cliquer sur Connect. Une fois connecté, vous aurez la structure de votre espace sur le serveur Etud sur la partie droite de votre fenêtre, et il vous suffit de glisser et déposer les fichiers souhaités de droite à gauche ou de gauche à droite. Pour les prochaines connexions il vous suffira de cliquer sur la flèche en haut à gauche et de sélectionner le serveur enregistré lors de la première connexion.

## Linux

Sur Linux vous pouvez directement passer par votre navigateur de fichiers.

Accès au serveur

# Accéder à sa base de donnée

Si vous avez créé votre base de donnée, il est très simple d'y accéder.

**Pas besoin de VPN**, seulement d'un navigateur web ! Pour cela, rendez vous sur [phpMyAdmin](#) et rentrez le login et mot de passe de votre base de donnée.

Si cette base de donnée est associée à un espace club, le login est celui du club, sinon c'est votre login INSA. Dans les deux cas, le mot de passe est différent de celui utilisé pour vous connecter à votre session, et vous a été donné par un Root lors de la création de la base de donnée.

# Utilisation du serveur

Comment utiliser le serveur pour créer un site web ou partager des fichiers

# Structure de votre espace web

Les fichiers à publier sur le site web doivent être placés dans un dossier `public_html`. Le serveur lis les pages avec un compte spécial (www-data) et par conséquent, pour rendre les pages accessibles sur internet, il faut qu'il puisse les lire. Vérifiez donc bien les droits associés à vos fichiers. Le plus simple est de donner le groupe www-data à tous vos fichiers et de donner le droit x au groupe.

Si vous souhaitez utiliser les variables de session en php, vous devrez créer au préalable un dossier "sessions" à la racine de votre espace (là où se trouve le dossier public\_html).

Utilisation du serveur

# Créer votre site web

Il existe de nombreux tutoriels sur internet pour créer des sites web.

# Systemes d'authentifications

Comment faire pour limiter l'accès à certaines page ? Comment faire pour authentifier des utilisateurs ? Les méthodes ci-dessous sont les seules autorisées par les règles du serveur Etud.

# Utiliser le CAS (Recommandé)

Le CAS est ce qu'on appelle un système de Single Sign-On (SSO). Il permet de centraliser l'authentification des utilisateurs ce qui permet d'éviter à devoir se reconnecter à tous les sites qui demandent une authentification. De plus cela permet d'améliorer la sécurité en évitant de devoir taper son mot de passe sur de multiples sites web. **C'est la méthode recommandée par le CSN, elle est donc à privilégier.**

## Utiliser le CAS dans un `.htaccess`

La méthode la plus simple pour restreindre l'accès à un page web avec l'authentification CAS est d'ajouter un fichier `.htaccess` à la racine du site. La configuration est la suivante :

```
AuthType Cas
AuthName "Authentication Required"
CASAuthNHeader CAS-User
Require valid-user
```

Lors d'une tentative d'accès au site, l'utilisateur sera alors redirigé vers le CAS pour s'authentifier.

# Utiliser l'authentification CAS en PHP

## Préparation

Avant toute chose, il vous faut autoriser les sessions pour votre espace personnel. Pour cela, vérifiez que vous possédez un dossier `sessions` à la base de votre dossier personnel (à coté de `public_html`).

## Installation

Téléchargez ensuite la dernière version stable de la librairie phpCAS comme indiqué sur [le guide d'installation](#), et extrayez cette archive dans votre dossier `public_html`. La liste de toutes les versions est [disponible ici](#). Il existe d'autres méthodes d'installation mais elles n'ont pas été testées sur Etud.

Par exemple pour la version 1.3.6, si vous êtes [connecté en SSH à votre espace club/étudiant](#), utilisez les commandes suivantes dans le dossier `public_html` :

```
wget https://github.com/apereo/phpCAS/archive/1.3.6.tar.gz
tar zxvf 1.3.6.tar.gz
```

Vous aurez ainsi les fichiers suivants dans votre dossier `public_html`, (affichés avec la commande `ls`) :

```
1.3.6.tar.gz  phpCAS-1.3.6
```

Vous pouvez supprimer le fichier `1.3.6.tar.gz` car il n'est plus nécessaire, et renommer le dossier `phpCAS-1.3.6` comme vous voulez.

L'installation est maintenant terminée !

## Utilisation

Voici un exemple d'utilisation. Si vous avez installé comme expliqué précédemment, créez un fichier `index.php` à côté du dossier de `phpCAS-1.3.6`. Le contenu du dossier `public_html` est alors le suivant :

```
index.php  phpCAS-1.3.6
```

Ouvrez le fichier `index.php` et ajoutez le contenu suivant (adapté de [cet exemple](#)) :

```
<?php
// Load the CAS lib
require_once("phpCAS-1.3.6/CAS.php");

// Enable debugging
phpCAS::setDebug();

// Enable verbose error messages. Disable in production!
phpCAS::setVerbose(true);

// Initialize phpCAS
```

```
phpCAS::client(CAS_VERSION_2_0, "cas.insa-toulouse.fr", 443, 'cas', true);

// For production use set the CA certificate that is the issuer of the cert
// on the CAS server and uncomment the line below
// phpCAS::setCasServerCACert($cas_server_ca_cert_path);

// For quick testing you can disable SSL validation of the CAS server.
// THIS SETTING IS NOT RECOMMENDED FOR PRODUCTION.
// VALIDATING THE CAS SERVER IS CRUCIAL TO THE SECURITY OF THE CAS PROTOCOL!
//phpCAS::setNoCasServerValidation();

// force CAS authentication
phpCAS::forceAuthentication();

// at this step, the user has been authenticated by the CAS server
// and the user's login name can be read with phpCAS::getUser().

// logout if desired
if (isset($_REQUEST['logout'])) {
    phpCAS::logout();
}

// for this test, simply print that the authentication was successfull
?>
<html>
  <head>
    <title>phpCAS simple client</title>
  </head>
  <body>
    <h1>Successfull Authentication!</h1>
    <p>the user's login is <b><?php echo phpCAS::getUser(); ?></b>.</p>
    <p>phpCAS version is <b><?php echo phpCAS::getVersion(); ?></b>.</p>
    <p><a href="?logout=">Logout</a></p>
  </body>
</html>
```

Voici une petite explication ligne par ligne de ce fichier :

```
require_once("phpCAS-1.3.6/CAS.php");
```

Cette ligne est essentielle ! Elle permet de charger la librairie CAS dans votre script PHP. Sans cette ligne vous ne pourrez pas utiliser les fonctions du CAS. Dans cet exemple, la librairie est dans le dossier `phpCAS-1.3.6`, mais si vous avez renommé ce fichier ou l'avez placé dans un autre dossier, entrez le chemin vers ce dossier à la place.

```
phpCAS::setDebug();
phpCAS::setVerbose(true);
```

**À supprimer pour un site en production !** Ces lignes sont utiles quand vous développez votre site pour voir les erreurs CAS. Quand vous pensez avoir un système robuste, pensez à enlever ces lignes.

```
phpCAS::client(CAS_VERSION_2_0, "cas.insa-toulouse.fr", 443, 'cas', true);
```

Cette ligne permet d'initialiser le CAS et le lier au serveur de l'INSA. Le dernier argument permet d'activer les sessions PHP (rappelez vous le dossier `sessions` que vous aviez créé au début).

```
phpCAS::setNoCasServerValidation();
```

Théoriquement, il faudrait pas utiliser cette ligne mais je sais pas où trouver le certificat du serveur de l'INSA donc bon.

```
phpCAS::forceAuthentication();
```

Cette ligne permet de n'afficher la page que si l'utilisateur est authentifié avec le CAS.

```
if (isset($_REQUEST['logout'])) {
    phpCAS::logout();
}
```

Ce bloc permet de déconnecter l'utilisateur si le paramètre `logout` est présent dans l'url. Par exemple, si votre site est à l'adresse `https://etud.insa-toulouse.fr/~CASTest`, alors `https://etud.insa-toulouse.fr/~CASTest?logout=` déclenchera la déconnexion.

```
<html>
  <head>
    <title>phpCAS simple client</title>
  </head>
  <body>
    <h1>Successfull Authentication!</h1>
    <p>the user's login is <b><?php echo phpCAS::getUser(); ?></b>.</p>
    <p>phpCAS version is <b><?php echo phpCAS::getVersion(); ?></b>.</p>
```

```
<p><a href="?logout=">Logout</a></p>
</body>
</html>
```

Ce bloc est le HTML affiché sur la page. Comme il ne sera affiché que si l'utilisateur est connecté, il est donc possible d'utiliser sans problème les méthodes de `phpCAS` pour récupérer des informations sur l'utilisateur. Ici, le nom d'utilisateur et la version de `phpCAS` sont récupérés avec les méthodes `phpCAS::getUser()` et `phpCAS::getVersion()`. Il y a aussi un lien déclenchant la déconnexion comme expliqué précédemment.

Pour des utilisations plus complexes, regardez [la liste de tous les exemples](#).

Si vous avez des problèmes, vous pouvez visiter [le code source sur github](#) et [le wiki de phpCAS](#).

## Laravel

Voici maintenant un exemple d'utilisation de l'authentification CAS avec le framework PHP [Laravel](#). Vous devez au préalable suivre l'installation via Composer comme indiqué en haut du [guide d'installation de phpCAS](#).

Dans le fichier `config/auth.php`, vous devez rajouter :

```
/*
|-----
| INSA CAS properties
|-----
|
| This option controls the INSA CAS authentication system used
| to log in a different way the users.
|
*/
'cas' => [
    'debug' => env('INSA_CAS_DEBUG', false),
    'server' => [
        'hostname' => env('INSA_CAS_HOSTNAME', ''),
        'port'     => (int) env('INSA_CAS_PORT', 443),
        'uri'      => env('INSA_CAS_URI', ''),
    ],
],
```

],

Dans le fichier `.env`, vous pouvez rajouter :

```
INSA_CAS_DEBUG=false
INSA_CAS_HOSTNAME=cas.insa-toulouse.fr
INSA_CAS_PORT=443
INSA_CAS_URI=cas
```

Le fichier `CAS.php` ci-dessous permet de regrouper dans un seul fichier l'utilisation de l'objet `phpCAS`.

```
<?php

namespace App\Traits;

use phpCAS;
use Illuminate\Http\RedirectResponse;

/**
 * Class CAS
 * @package App\Traits
 *
 * @author: Damien MOLINA
 */
class CAS {

    /**
     * Determine whether the CAS auth system is currently
     * in a debugging mode.
     *
     * @return bool
     */
    private static function isDebuggingMode() {
        return boolval(
            config('auth.cas.debug')
        );
    }

    /**
```

```

    * Initialize the CAS object.
    *
    * @return void
    */
private static function initialize() {
    if(CAS::isDebuggingMode()) {
        // Enable debugging.
        phpCAS::setDebug();

        // Enable verbose error messages.
        phpCAS::setVerbose(true);
    }

    phpCAS::client(
        CAS_VERSION_2_0, config('auth.cas.server.hostname'), config('auth.cas.server.port'),
        config('auth.cas.server.uri'), true
    );
}

/**
 * Make the authenticate request.
 *
 * @param string $returnRoute
 */
private static function authenticate(string $returnRoute) {
    // We initialize the phpCAS component
    CAS::initialize();

    // We put the return URL.
    phpCAS::setFixedServiceURL($returnRoute);

    // Disable the CAS server verification. UNCOMMENT TO DEBUG.
    //phpCAS::setNoCasServerValidation();

    // force CAS authentication
    phpCAS::forceAuthentication();
}

/**
 * Make the authentication request.

```

```

    *
    * @param string $returnRoute
    * @return RedirectResponse|void
    */
    public static function request(string $returnRoute) {
        CAS::authenticate($returnRoute) ;

        /**
         * Now we are redirecting the user
         * to the login page.
         */
        return redirect()->to(
            phpCAS::getServerLoginURL()
        ) ;
    }

    /**
     * Get all the data from the request.
     *
     * @param string $returnRoute
     * @return array
     */
    public static function response(string $returnRoute) {
        /**
         * We make the request again to take
         * the server data without redirecting
         * the user.
         */
        CAS::authenticate($returnRoute) ;

        return [
            'login' => phpCAS::getUser(),
            'version' => phpCAS::getVersion(),
        ] ;
    }
}

```

Vous pourrez ainsi utiliser l'objet `CAS` dans un contrôleur en définissant, par exemple, les méthodes `CASRequest` et `CASResponse` :

```

/**
 * Make the CAS authentication request and
 * redirect the user to the login page.
 *
 * @return RedirectResponse
 */
public function CASRequest() {
    return CAS::request(
        $this->getCasReturnRoute()
    );
}

```

```

/**
 * Manage the user return.
 *
 * @return RedirectResponse
 */
public function CASResponse() {
    $data = CAS::response(
        $this->getCasReturnRoute()
    );

    $login = $data['login'];

    // Do something with the login
}

```

```

/**
 * Get the CAS return route.
 *
 * @return string
 */
private function getCasReturnRoute() {
    return route('your-route');
}

```

Dans ce cas, la variable `$login` contiendra l'identifiant utilisé dans le service CAS.

# Utiliser l'authentification CAS dans d'autres langages

Le CAS propose des clients dans d'autres langages que le PHP. Ces utilisations ne sont pas couvertes par ce wiki car peu utilisées sur le serveur Etud. Si vous voulez tout de même utiliser l'un de ces clients, vous pouvez visiter le [wiki du CAS](#).

# Utiliser un .htaccess

Un .htaccess permet de limiter l'accès à un dossier, soit totalement, soit par un mot de passe. C'est une méthode très simple et fiable, qui peut être utilisée pour faire un espace administrateur basique. Pour l'utiliser, créez simplement un fichier nommé `.htaccess` dans le dossier que vous voulez protéger, puis suivez les instructions adaptées à votre besoin ci-dessous.

## Identifiant et mot de passe personnalisé

Vous allez devoir créer des couples identifiant/mot de passe et les stocker dans un fichier. Ce seront les seuls comptes autorisés à accéder au dossier protégé.

### Création des comptes autorisés

Pour cela, créez tout d'abord un fichier nommé `.htpasswd` dans le dossier de votre choix (si possible dans un emplacement sécurisé). Ensuite, allez sur [ce site](#) pour chiffrer votre mot de passe.

Sur ce site, entrez l'identifiant souhaité et le mot de passe associé, choisissez le mode Bcrypt, vous obtiendrez alors une ligne du type `login:mot de passe chiffré`. Copiez cette ligne dans votre fichier `.htpasswd`.

```
coucou:$2y$10$/1/DjZ/3kwou4PcrMjPzE000/YMXFIMAvZcZQSWA23Jla8DWRoHnO
```

Username

Enter the username you would like to add your .htpasswd file.

Password

Enter the password to be encrypted.

Mode

\* Bcrypt (Apache v2.4 onwards)

Create .htpasswd file

Clear

Vous pouvez créer autant de couples identifiant/mot de passe que vous voulez, et les stocker dans le même `.htpasswd`, ligne par ligne.

# Écriture du .htaccess

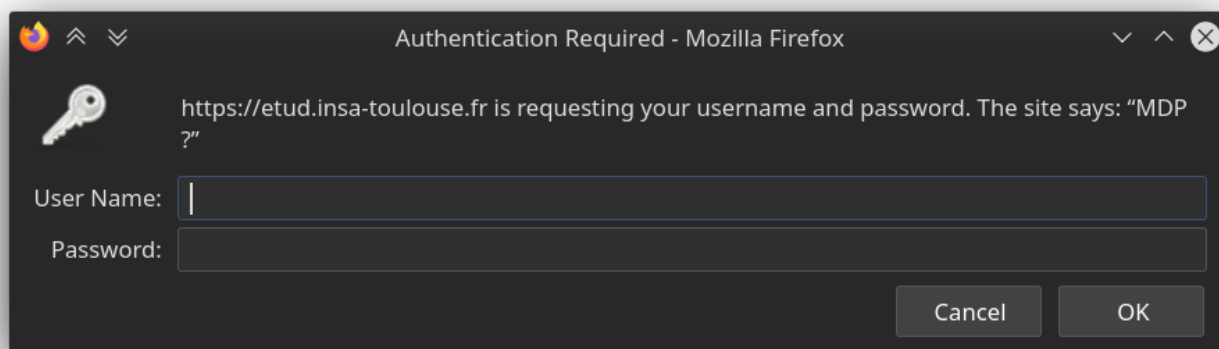
Dans le fichier `.htaccess` que vous venez de créer, rentrez les informations suivantes :

```
AuthUserFile /home/{LOGIN}/public_html/{CHEMIN}/.htpasswd
AuthName "Accès réservé"
AuthType Basic
Require valid-user
```

Sur la ligne `AuthUserFile`, remplacez les choses suivantes :

- `home` : Si vous êtes dans un espace club, remplacez par `home_club`, sinon laissez tel quel.
- `{LOGIN}` : Remplacez par votre login INSA, ou celui de votre club
- `{CHEMIN}` : Remplacez par le chemin vers votre `.htpasswd`, relatif au dossier `public_html`

Votre dossier est maintenant accessible seulement par les comptes que vous avez autorisé, et vous devrez avoir une fenêtre vous demandant d'entrer vos identifiants comme ci-dessous:



## Identifiant et mot de passe INSA Avec le CAS (Recommandé)

Voir la [page dédiée](#).

## Avec le LDAP (Déprécié)

Si vous voulez seulement restreindre l'accès aux étudiants INSA, merci de suivre [la procédure décrite sur la page du LDAP](#).

# Utiliser le LDAP

⚠ L'utilisation du LDAP est dépréciée, nous vous invitons fortement à basculer vers l'authentification CAS. Les serveurs LDAP ci-dessous seront bientôt décommissionnés.

---

Le LDAP est un protocole permettant d'authentifier les utilisateurs membre d'un annuaire. L'INSA met à disposition son propre LDAP pour authentifier les étudiants.

⚠ Pour des raisons évidentes de sécurité, le LDAP de l'INSA n'est accessible que depuis le serveur étudiant. Vous ne pourrez donc pas le tester sur votre machine.

## Configuration utilisée

Voici la configuration pour se connecter au LDAP de l'INSA. Veuillez noter que l'INSA utilise maintenant le **LDAPS**, qui est une version sécurisée du LDAP. Le port utilisé est ainsi différent de celui utilisé habituellement et que vous pouvez trouver sur internet. La configuration ci-dessous peut vous être utile si vous souhaitez installer un site compatible avec cette méthode d'authentification.

```
Host: srv-ldap1.insa-toulouse.fr
Port: 636
User Search Base: ou=People,dc=insa-toulouse,dc=fr
User Filter: (&(eduPersonAffiliation=student)(uid=%s))
Username: uid
First Name: givenName
Last Name: sn
Email: mail
```

Utiliser avec un `.htaccess`

Utiliser le LDAP avec un `.htaccess` est très simple et sécurisé. Pour commencer, merci de suivre les instructions sur [cette page](#).

Maintenant que vous avez votre fichier `.htaccess`, ouvrez le et rentrez les informations suivantes :

```
AuthName "Entrez vos identifiants INSA pour continuer"  
AuthType Basic  
AuthLDAPURL ldaps://srv-ldap1.insa-toulouse.fr:636/ou=People,dc=insa-toulouse,dc=fr  
AuthBasicProvider ldap  
  
require valid-user
```

Votre dossier est maintenant accessible seulement par les membres de l'INSA.

## Utiliser avec PHP

**Merci de ne pas utiliser le LDAP avec PHP.** Préférez le CAS qui est plus sécurisé et puissant que le LDAP.

# Astuces

# Changer son mot de passe

Il est intéressant de changer régulièrement de mot de passe, lors par exemple d'une passation de club. Cela permet de limiter des accès non autorisés à vos données.

Pour ceux qui veulent changer leur mot de passe INSA, rendez-vous sur la page <https://wiki.etud.insa-toulouse.fr/books/astuces-ressources/page/changer-son-mot-de-passe-insa>

## Base de donnée

Que ce soit pour un espace club ou étudiant, il faut demander aux Roots Étudiants par mail ([root-etud@insa-toulouse.fr](mailto:root-etud@insa-toulouse.fr)) de changer le mot de passe de votre base de donnée. Pour des raisons de sécurité, personne sauf les Roots Étudiants ne peuvent changer les mots de passe des bases de données.

## Connexion à l'espace

### Pour les clubs

Un club est un utilisateur standard Unix, il est donc possible de changer son mot de passe comme habituellement sous Unix. Si vous n'avez rien compris à cette phrase, vous pouvez tout simplement demander aux gentils Roots Étudiants de le changer pour vous.

### Pour les étudiants

Le mot de passe de connexion à votre espace est celui de votre compte INSA. Vous pouvez le changer à tout moment sur le site des comptes de l'INSA en vous connectant à votre VPN.